The optimisation framework Geneva in science: Using massively parallel processing to solve complex optimisation problems

Kilian Schwarz / Matthias Lutz
Rüdiger Berlich / Ariel Garcia / Jan Knedlik
Denis Bertini / Jannis Geuppert

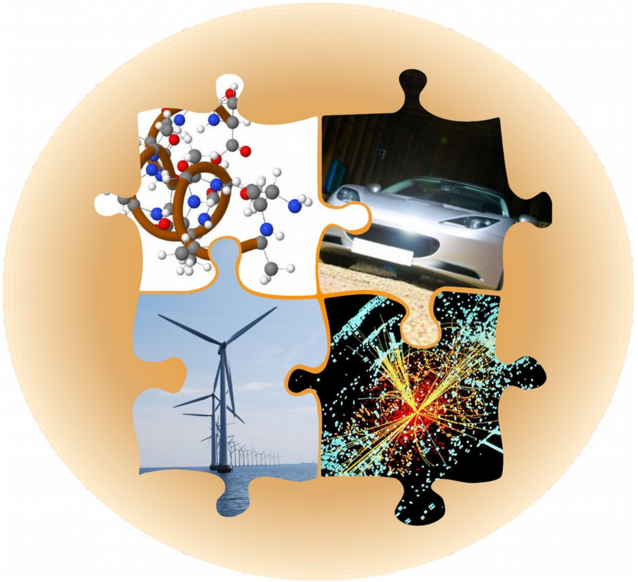
GSI Helmholtzzentrum für Schwerionenforschung GmbH
Gemfony scientific UG (haftungsbeschränkt)

03.02.2021 / 7th MT Meeting

Contact:
k.schwarz@gsi.de
r.berlich@gemfony.eu

The Geneva Library Collection

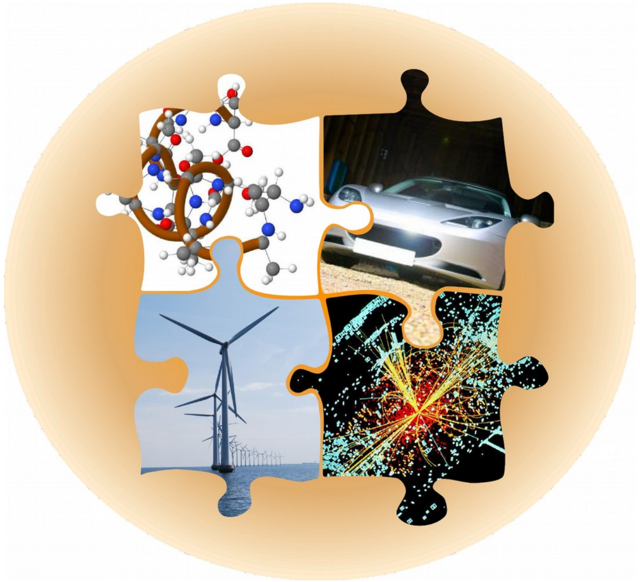
- Generic C++ framework for the search for **optimized solutions** of technical and scientific problems
- Covering **Evolutionary Algorithms**, Swarm Algorithms, Simulated Annealing, Parameter Scans and Gradient Descents
- Data structures allow direct interaction between different optimization algorithms with **just one problem description**
- **Inter-parameter constraints (x+y < 1) possible**
- Support for **many-core systems** as well as parallel and distributed environments
- Available under the Apache License v2 (get it from https://github.com/gemfony/geneva)



Picture credits: see last page

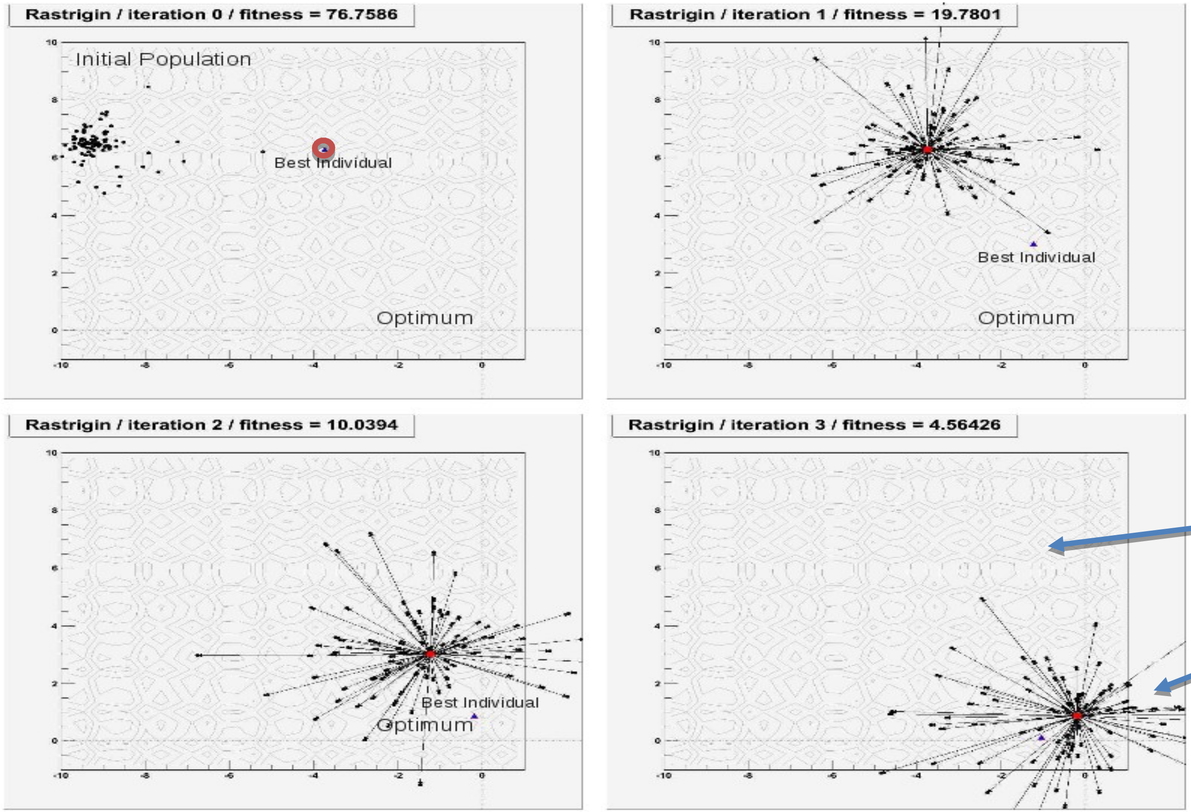
Our Vision

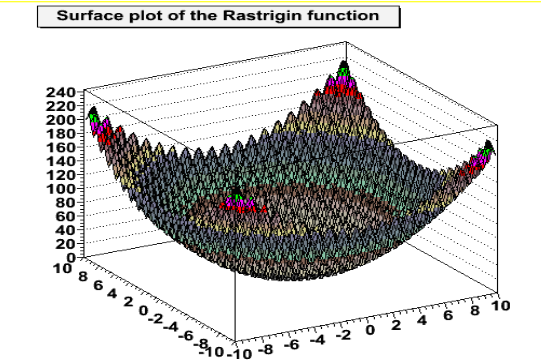
- To create a common Open Source optimization framework, constantly developed and extended by physicists, computer scientists and software engineers according to professional standards, covering the most recent algorithms for massively parallel optimization studies in research and industry
- To concentrate, activate and leverage fragmented knowledge in the field of parametric optimization to enable each other solving even the most daunting optimization problems



Picture credits: see last page

Dealing with Complex Quality Surfaces – Example: Evolutionary Algorithms





View in –z direction

Parent-individual (red) and children

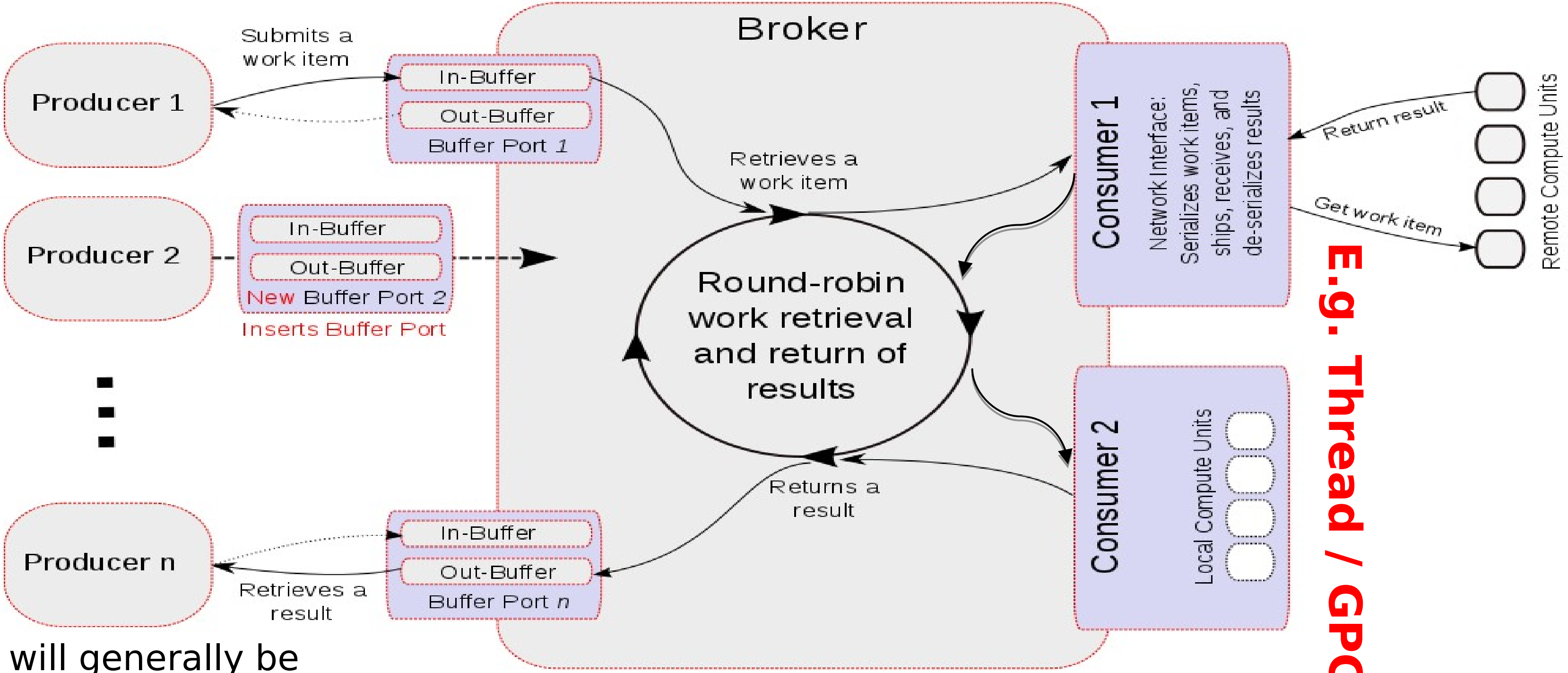
Source: Gemfony

GSI Helmholtzzentrum für Schwerionenforschung GmbH

Gemfony scientific

Parallelisation: more complex in distributed environments
A brokered multi-producer/multi-consumer architecture

Source: Gemfony



E.g. Thread / GPGPU

This will generally be
An optimization algorithm

Producers write work items through a broker.
Consumers read and process these work items
Plugabble consumers communicate with the spawned network-clients

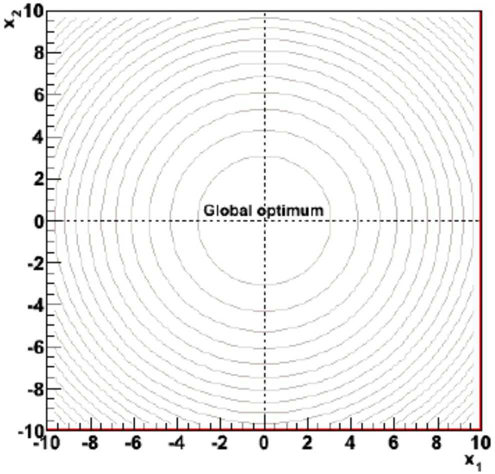
GSI Helmholtzzentrum für Schwerionenforschung GmbH

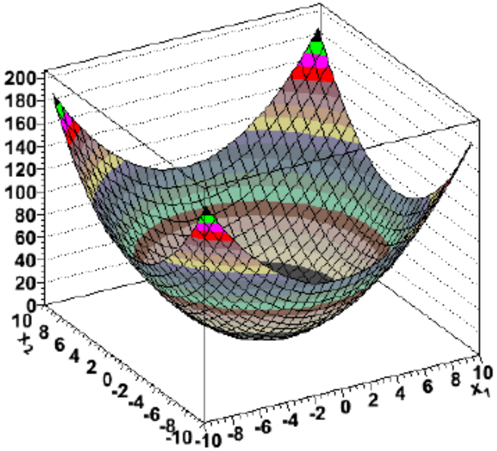
Gemfony scientific

Manual see https://www.gemfony.eu/fileadmin/documentation/geneva-manual.pdf

- Defining a first optimisation problem
- In an n-dimensional paraboloid, the „quality“ of the parameter set (n floating point numbers in this case) is defined as follows:

$$f(x_1,x_2,\ldots,x_n) = \sum_{i=1}^{n} x_i^2 = x_1^2 + x_2^2 + \ldots + x_n^2$$





Class GParaboloidIndividual2D

```
class GParaboloidIndividual2D :public GParameterSet
{
public:
  GParaboloidIndividual2D(); // default constructor
  GParaboloidIndividual2D(const GParaboloidIndividual2D&); // copy constructor
  virtual ~GParaboloidIndividual2D(); // destructor

protected:
  // Loads the data of another GParaboloidIndividual2D
  virtual void load_(const GObject*);
  // Creates a deep clone of this object
  virtual GObject* clone_() const;

  // Calculates the object's quality
  virtual double fitnessCalculation();

private:
  // Make the class accessible to Boost.Serialization
  friend class boost::serialization::access;

  // Triggers serialization of this class and its base classes.
  template<typename Archive>
  void serialize(Archive & ar, const unsigned int) {
    using boost::serialization::make_nvp;
    // Serialize the base class
    ar & BOOST_SERIALIZATION_BASE_OBJECT_NVP(GParameterSet);
    // Add other variables here like this:
    // ar & BOOST_SERIALIZATION_NVP(sampleVariable);
  }

  const double PAR_MIN_; // Lower boundary for parameters
  const double PAR_MAX_; // Upper boundary for parameters
};
```

The fitness calculation

```
double GParaboloidIndividual2D::fitnessCalculation(){
  double result = 0.; // Will hold the result
  std::vector<double> parVec; // Will hold the parameters

  this->streamline(parVec); // Retrieve the parameters

  // Do the actual calculation
  for(std::size_t i=0; i<parVec.size(); i++) {
          result += parVec[i]*parVec[i];
  }

  return result;
}
```

First output

```
Seeding has started
Starting an optimization run with algorithm "Evolutionary Algorithm"
0: 64.6073443050163
1: 25.9597623490252
2: 8.89715425355864
3: 1.45564799125829
4: 0.861887897798893
[...]
999: 7.37074272148514e-13
End of optimization reached in algorithm "Evolutionary Algorithm"
Done ...
```

In the Client Server mode many clients/individuals can run in parallel and contribute to solving a complex problem

Hadronic reaction amplitudes for FAIR

- A framework for predicting and analysing final-state interactions for the FAIR experiments is beeing developed
- This requires massive parallel computing, up to 50 and more coupled-channels needed
- Reaction amplitudes are derived from effective Lagrangians where coupled-channel unitarity and the implications of micro-causality (dispersion relations) are implemented (isobar models are not good enough)
- Parameter space is reduced significantly by using constraints from chiral and heavy-quark symmetry but also large-Nc QCD
- A subset of the parameters can be derived from the quark-mass dependence of existing QCD lattice data and/or fits to existing data
- Conventional fitting routines like Minuit are not suitable for such problems – gradients are expensive and not stable
- In order to avoid local minima and to be able to find the best possible solution an Evolutionary Algorithm with reasonably high population has proven to be effective

Projects done with Geneva

Constraints from a large-Nc analysis on meson-baryon interactions at chiral order Q3
Y. Heo, C. Kobdaj, M.F.M. Lutz
Published in: Phys. Rev. D 100 (2019) 9, 094035

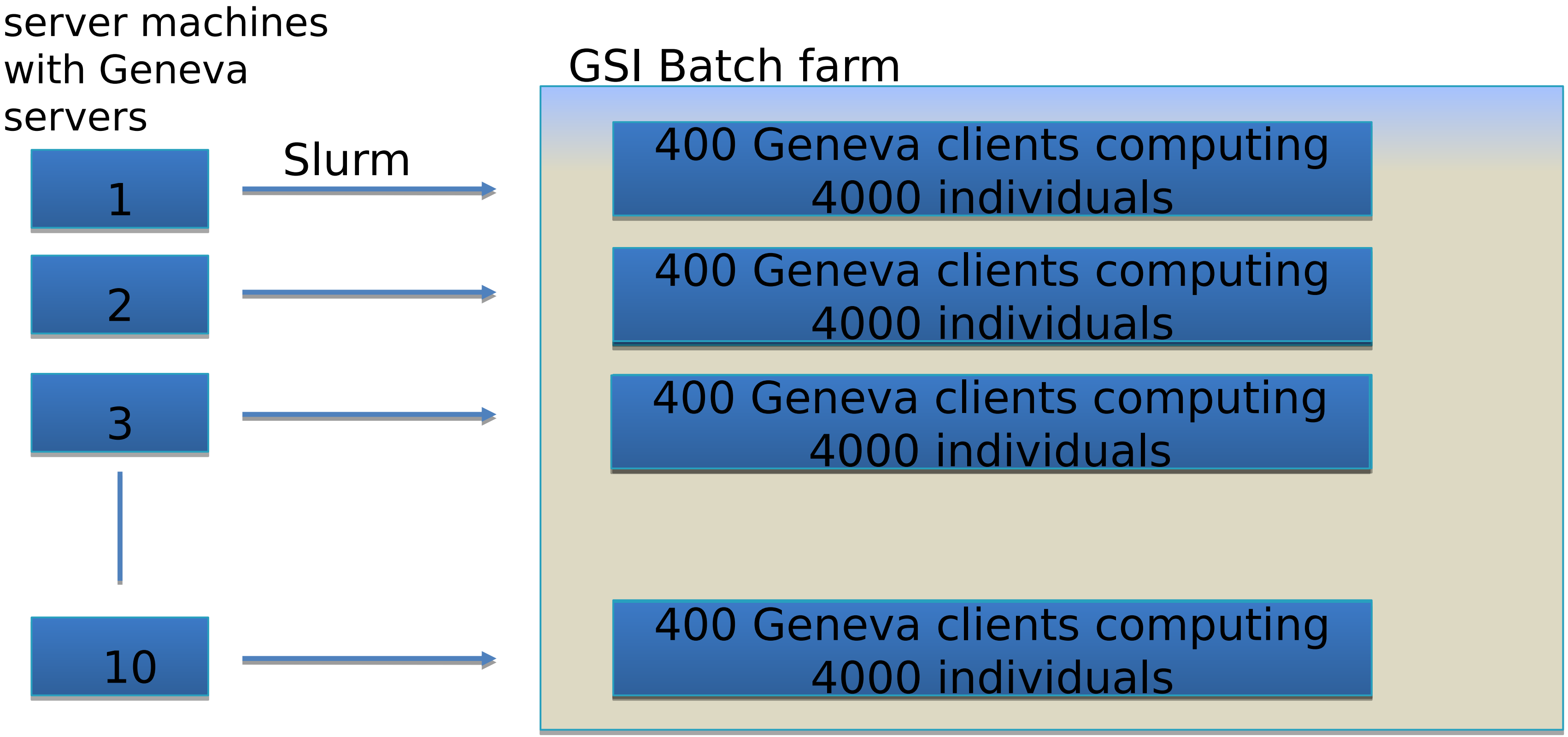
On a first order transition in QCD with up, down, and strange quarks
Xiao-Yu Guo, Y. Heo, M.F.M. Lutz
Published in: Eur. Phys. J. C 80 (2020) 3, 260

A generalised Higgs potential with two degenerate minima for a dark QCD matter scenario
M.F.M. Lutz, Y. Heo, Xiao-Yu Guo
Published in: Eur. Phys. J. C 80 (2020) 4, 322

From Hadrons at Unphysical Quark Masses to Coupled-Channel Reaction Dynamics in the Laboratory
M.F.M. Lutz, Xiao-Yu Guo, Y. Heo
Published in: JPS Conf. Proc. 26 (2019) 022022

Low-energy constants from charmed baryons on QCD lattices
Y. Heo, Xiao-Yu Guo, M.F.M. Lutz
Published in: Phys. Rev. D 101 (2020) 5, 054506

Geneva Cluster @ GSI
example case: 10 minutes compute time for one solution, 10 x 400 clients, 10 x 4000 population, 1000 iterations, one week of compute time in total



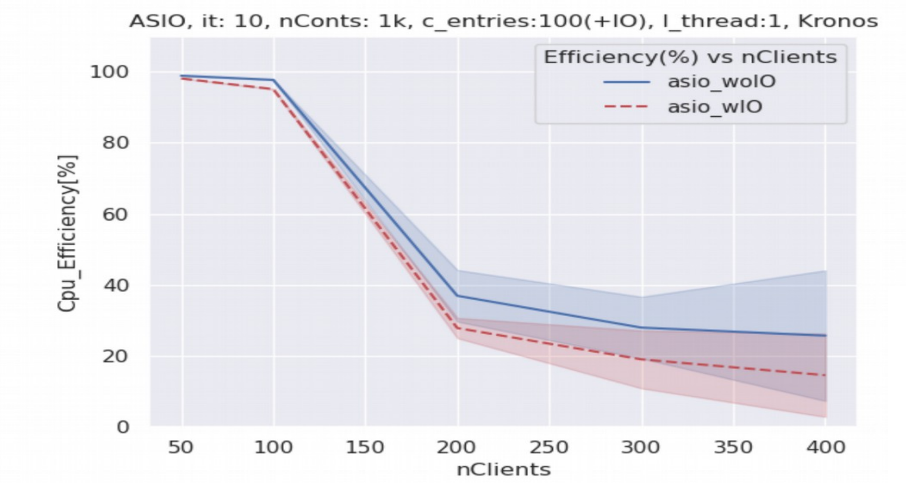
- Geneva client-server communication
  - transition from Boost.ASIO-Sockets to Beast Websockets
  - heartbeat option allows better client control
  - less load on server, higher number of clients, higher efficiency
- Checkpoints
  - iterations are stored in checkpoints (text, xml or binary)
  - iterations can be continued later by loading the checkpoint file
- Geneva Testing Suite, MPI Consumer, Python Interface

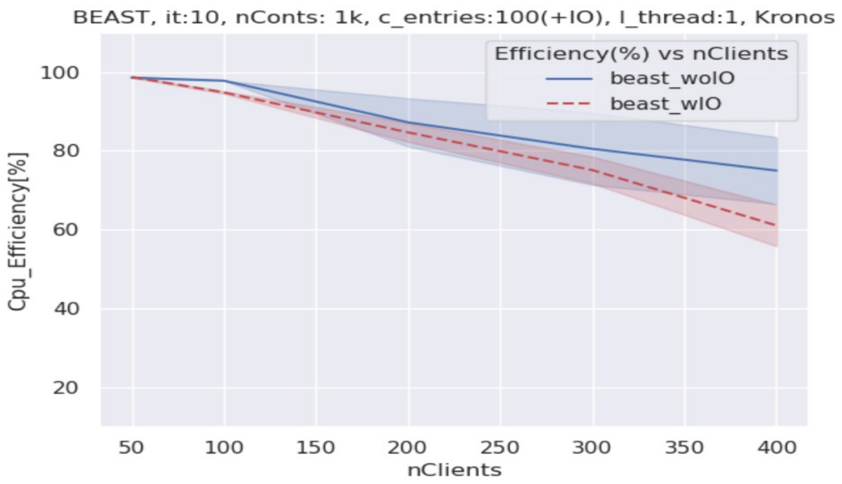
Benchmarks CPU Efficiency vs. nClients

- ongoing developments based on Geneva benchmark suite
- added values
  - Beast WebSocket Consumer/Client added
  - Container classes (Simple, Random) with specific work load (in function process())
  - useful to test changes in the Geneva code base
- automatic script for starting a job in the Cluster

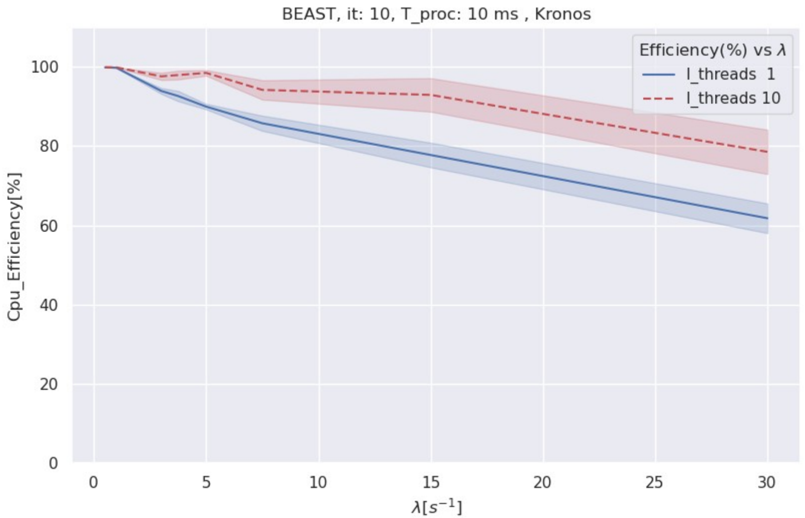
```
dbertini@lxbk0198:/lustre/rz/dbertini/ea/perfs$ ./tracejob.rb -h
Usage: tracejob [options]
    -p, --producers PRODUCERS        number of producers (default 1)
    -w, --workers WORKERS            number of workers (default 1)
    -c, --prod_cycles PROD_CYCLES    number of production cycles (default 2)
    -o, --cont_objs CONT_OBJS        number of container objects (default 10)
    -e, --cont_entries CONT_ENTRIES  number of container entries (default 10)
    -t, --l_threads LISTTHREADS      number of listener threads (default 1)
    -x, --ipc IPC                    networking strategie (default 9)
    -s, --nclients NCLIENTS          number of clients jobs (default 2)
    -q, --queue QUEUE                cluster queue (default main)
    -g, --io IOTEST                  Container IO (default 0)
    -h, --help                       Display this screen
```

ASIO vs. Beast CPU Efficiency and number of clients

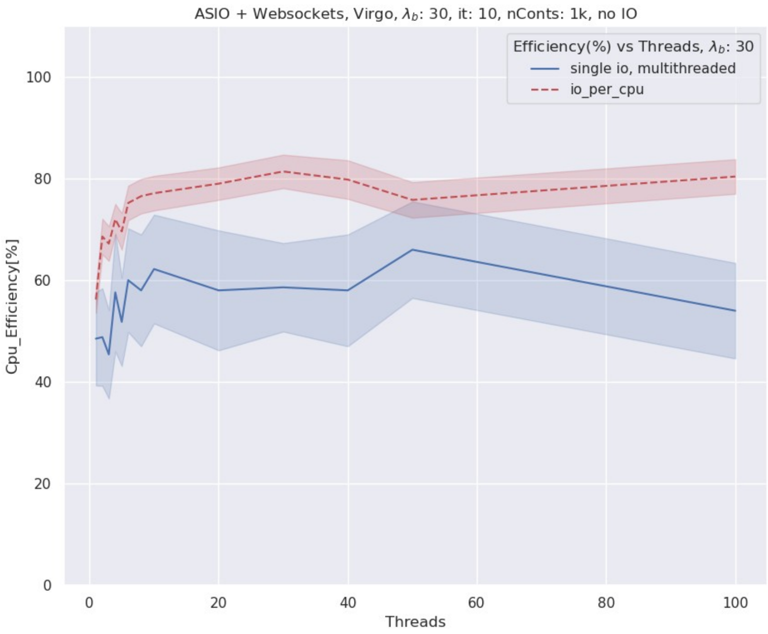




Even further efficiency improvements

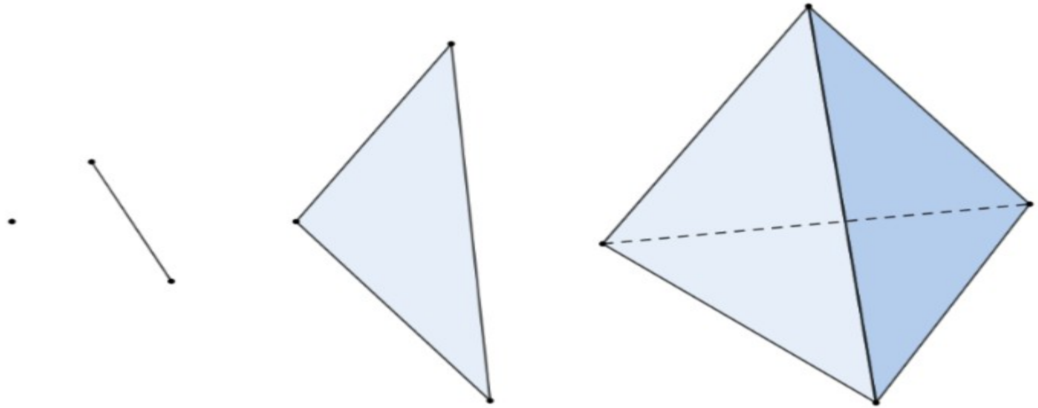


Optimising
lambda_b = N_clients/[N_cont_objs * Proc_time]
and several listener threads per consumer



New consumer being multi-threaded with several io_context instances and several threads

- Implementation of a parallel Nelder-Mead-Simplex algorithm
- Based on the geometrical form of the Simplex
- Planned to be included in Geneva



Conclusion

- Geneva is an efficient client/server tool for doing distributed optimisation within an HPC environment
  - mainly using Evolutionary Algorithm
  - using up to 400 clients per server dealing with population up to 4000
- May need refactoring and needs a larger community
  - Needs ideas both for optimization and for the clustering part
- Future work
  - adding more reliable optimisation algorithms
  - increasing scalability
  - starting inter-site optimisation on Grids/Clouds
  - Geneva Spack package and Geneva Singularity Container for easy use
  - Simplify Geneva interface even more for common usage patterns

Thank you

Do contact us in case of questions:

k.schwarz@gsi.de

r.berlich@gemfony.eu

If you want to try Geneva:

https://github.com/gemfony/geneva

http://www.gemfony.eu

Masthead Gemfony

Address	Gemfony scientific UG (haftungsbeschränkt) Hauptstraße 2 76344 Eggenstein-Leopoldshafen - Germany
Telefone	+49(0)7247/934278-0
Fax	+49 (0)7247 934 2781
Email	
Registered at	Amtsgericht Mannheim (Germany)
Registration-Id	HRB 710566
Ust.-Id	DE274421406
Managing Director	Dr. Rüdiger Berlich

Material used

Slide	Figure	Source
The Geneva library collection	Car in puzzle	Image courtesy of Simon Howden at FreeDigitalPhotos.net
The Geneva library collection	Wind turbines in puzzle	http://www.flickr.com/photos/pebondestad/3533177131/sizes/l/in/photostream/Creative Commons Attribution 2.0; By Pål Espen Bondestad.
The Geneva library collection	Particle decay	https://en.wikipedia.org/wiki/File:CMS_Higgs-event.jpg ; Creative Commons Attribution Share-Alike 3.0; By CERN
Other slides	Other pictures	Gemfony scientific + GSI